# Semantic Inference from Natural Language Privacy Policies and Android Code

Mitra Bokaei Hosseini
Department of Computer Science, University of Texas at San Antonio
San Antonio, Texas, United States
mitra.bokaeihosseini@utsa.edu

## ABSTRACT

Mobile apps collect different categories of personal information to provide users with various services. Companies use privacy policies containing critical requirements to inform users about their data practices. With the growing access to personal information and the scale of mobile app deployment, traceability of links between privacy policy requirements and app code is increasingly important. Automated traceability can be achieved using natural language processing and code analysis techniques. However, such techniques must address two main challenges: ambiguity in privacy policy terminology and unbounded information types provided by users through input fields in GUI. In this work, we propose approaches to interpret abstract terms in privacy policies, identify information types in Android layout code, and create a mapping between them using natural language processing techniques.

## CCS CONCEPTS

• **Software and its engineering** → **Requirements analysis**; • **Security and privacy** → *Usability in security and privacy*; • **Computing methodologies** → *Information extraction*;

## KEYWORDS

Requirements Engineering, Privacy, Traceability, natural Language Processing

**ACM Reference Format:**
Mitra Bokaei Hosseini. 2018. Semantic Inference from Natural Language Privacy Policies and Android Code. In *Proceedings of the 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '18), November 4–9, 2018, Lake Buena Vista, FL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3236024.3275427

## 1 INTRODUCTION

Mobile and web applications (apps) are increasingly popular due to the convenient services they provide in different domains of interest. With growing access to personal information and the scale of mobile app deployment, the need for tools to help developers to protect user privacy is increasingly important. Regulators [5, 7, 12] require apps to provide users with a legal privacy notice, also called a privacy policy, which can be accessed by users before installing the app. Privacy policies contain critical *requirements* that inform stakeholders about data practices [1]. Mobile app developers can improve accountability and demonstrate that their apps comply with their policies by maintaining trace links between the policy requirements and their app code. To conveniently identify the trace links as the code evolves, an automated technique is required to extract and organize the information types described in *policies* and *code*.

In privacy policies, data practices are commonly described using hypernymy [3], which occurs when a more abstract information type is used instead of a more specific information type. Hypernymy permits multiple interpretations of words, which can lead to inconsistency in traceability. To illustrate, we draw an example from Adobe's privacy policy stating that "when you activate your Adobe product, we collect certain *information about your device*, the Adobe product, and your product serial number." This statement mentions the collection of the abstract information type "information about your device" that can be interpreted in various ways. For example, since a mobile device is a kind of device, we can infer that the statement also implies the collection of "mobile device information", which may include "device IP address." These interpretations arise when the hypernym "mobile device information" is recognized by a person, such as a developer, and matched with phenomena in the world based on their experience and background knowledge (e.g, by calling a platform API method in the mobile app code).

In app code, prior work by Slavin et al. [23] and Zimmeck et al. [32] attempt to identify information types collected through API method calls with static analysis. These API method calls concern personal information that are automatically collected from the device, such as sensor data. These works are not focused on addressing personal information that *users provide directly through user interface (UI).* The information types automatically collected through platform API methods are constrained to Android APIs which are described by comprehensive documents and information collected is well defined. These constraints limit the terminological space to only a few general category names (e.g., location, voice, etc.) In contrast, developers can design novel UIs that ask users to provide potentially *any kind of information*, which includes unstructured and semi-structured personal information in different formats and language types. Figure 1 shows an example where sensitive information is provided to the app via the interface and is thus disconnected from any API method call. These user-based input fields are difficult to identify as they are both context-sensitive and can vary in implementation from developer to developer. Wang et al. [27] stress the sensitivity of user input information types

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*ESEC/FSE '18, November 4–9, 2018, Lake Buena Vista, FL, USA*
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5573-5/18/11…$15.00
https://doi.org/10.1145/3236024.3275427



RIGHTSLINK

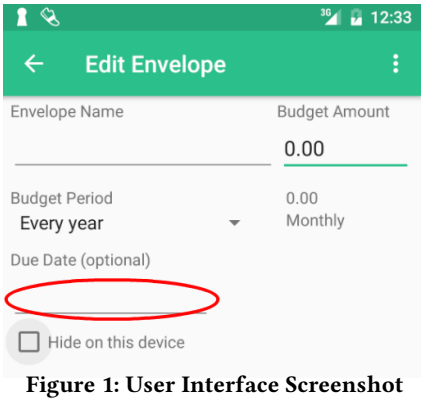


**Figure 1: User Interface Screenshot**

by analyzing 120 apps and identifying 39 inconsistencies between privacy policies and mobile apps leading to user input information type leakage.

In Android framework, static layout files contain the structure of pre-drawn UIs, view IDs, and all the text labels we can see from the UIs. Just like natural language text, input field views can only be well understood with neighboring/ancestor views. For the circled input field in Figure 1, without considering the context "envelope" only "due date" can be inferred as the information type. If the privacy policy contains the collection of "bill information" or "envelope information", the automatic consistency checkers fail to trace "due date" to "envelope information" without further context information. Therefore, UI context is essential in understanding user input information types.

**Problem Statement:** Abstract and ambiguous information type phrases in privacy policies, along with vague and unbounded information types for user input data are the main technical challenges in identifying the trace links between privacy policy requirements and app code.

**Thesis statement:** To address these challenges, this thesis analyzes three subject matters: (1) privacy policy ontologies that formalize multiple interpretations of the natural language information types described as being accessed, collected, or shared; (2) an approach to automatically identify information types associated with user input fields using sequence to sequence modeling; (3) a mapping to trace information types from Android layout code to privacy policy terminology considering abstraction and ambiguity.

## 2 BACKGROUND AND RELATED WORK

In this section, we provide definitions for some of the relations that hold among natural language phrases, and also review some related work.

Hypernymy is a semantic relation that the superordinate term (hypernym) extensionally subsumes subordinate term (hyponym). Meronymy is another semantic relation representing part-whole between meronym and holonym terms. Synonymy is a semantic relation between two terms when the meaning of two senses of two terms are identical or nearly identical. Lexicon is a collection of phrases or concept names that may be used in an ontology. Ontology is a collection of concept names and semantic relations between these concepts, including hypernymy, meronymy, and synonymy, among others.

### 2.1 Lexical Ontology

WordNet is a lexical database that contains English words grouped into nouns, verbs, adjectives, adverbs, and function words [9, 17]. Within each category, the words are organized by their semantic relations, including hypernymy, meronymy, and synonymy [9]. Our analysis shows that only 14% of privacy-related information types are found in WordNet, mainly because the privacy policy lexicon is populated with multi-word, domain-specific phrases [14]. This analysis signifies the importance of creating formal ontologies for privacy policy domain.

### 2.2 Relationship Extraction and Classification

Snow et al. [24] presented a machine learning approach using hypernym-hyponym pairs in WordNet to identify additional pairs in the parsed sentences of the Newswire corpus. This approach relies on the explicit expression of hypernymy pairs in text. Marti Hearst proposed six lexico-syntactic patterns to automatically identify hypernymy in text using noun phrases and regular expressions [13]. Evans et al. [8] applied an extended set of 72 Hearst patterns to privacy policies to extract hypernymy pairs. However, pattern sets are limited because they must be manually extended to address new policies. There are feature-based and neural network models used to extract the relationships between annotated nominals in a given sentence [4, 10, 18, 29–31]. These approaches are sentence dependent and fail to consider the relations between phrases that are not in the same sentence. Therefore, our proposed work aims to model the semantic relations of two information types extracted from a pool of privacy policies.

### 2.3 Mapping from Privacy Policy to App Code

Slavin et al. [23] manually constructed a mapping from API invocations to policy phrases in the ontology. Wang et al. [27] created a mapping between UI labels and ontology concepts using WordNet similarity. Since WordNet calculates similarity only for word pairs, they extended it to map phrase pairs by simple greedy alignment. We plan to identify the mapping between two phrases using a relation classifier which utilizes word embeddings trained on privacy policy domain. Therefore, we believe our proposed mapping method can outperform the mapping proposed by Wang et al. [27].

### 2.4 Privacy Policy Requirements Analysis

Petronella presents a tool that relates natural language privacy policy statements to Android permissions [19] using a lookup table that includes a mapping from a pre-defined set of information types to Android permissions. Zimmeck et al. [32] attempt to identify the trace links between privacy and app code with respect to device ID, location, and contact information as a pre-defined set. Harkous et al. [11] introduces a framework to visualize the flow of user data being collected by mapping privacy policy segments to information types from a pre-defined set [28]. The above techniques fail to extract the exact information type from the policies and increases the abstraction level through the mapping, which leads to ambiguity. These works also fail to consider the introduction of new information types which don't exist in the pre-defined sets.

## 3 PRELIMINARY WORK

In this section, we briefly discuss the preliminary work to address the challenges mentioned in Section 1.



RIGHTSLINK

### 3.1 Manual Ontology Construction Approach

We developed a manual approach to construct a formal ontology that explicitly states what kinds of information are included in the interpretations of data-related concepts [16]. We applied content analysis, which is a qualitative research method for annotating text to identify words and phrases that embody the meaning of special codes [22], and grounded theory [6] to discover seven heuristics for manually classifying information types into a formal ontology. We evaluated these heuristics on 351 information types extracted from 50 mobile app privacy policies [16]. Slavin et al. [23] and Wang et al. [27] utilized this approach to construct formal ontologies on privacy policies to detect inconsistencies between privacy policies and Android mobile app code. These ontologies are published publicly [1] [2].

This approach requires at least two analysts comparing each information type with every other information type in the privacy policy lexicon, and assigning a semantic relationship to each pair. Considering the number of information types in a lexicon, this approach lacks scalability. To address this problem, we developed a semi-automated semantic analysis method discussed in Section 3.2.

### 3.2 Semi-automatic Ontology Construction

The manual ontology construction approach discussed in Section 3.1 requires paired comparison of $n*(n-1)/2$ for $n$ phrases in a lexicon. To overcome this problem, we developed a set of 17 initial semantic rules that are automatically applied to information types yielding a subset of all possible relations [14]. To improve the approach, we established a training set by asking human subjects to perform the more time-consuming task of comparing information types in the privacy policy lexicon containing 351 information types. For this reason, we constructed 2,365 phrase pairs that share at least one word, since the prospects produced by the semantic rules all share at least one common word. We then compared the results of the semantic rules against these human interpretations, which led to identifying 9 additional semantic rules. Finally, we evaluated the improved semantic rules using 109 unique information types extracted from six privacy policies, and human subject surveys to measure the correctness of the results produced by the semantic rules [14]. The results reveal that the method scales by reducing the paired comparisons by 74% and produces correct relations with a 1.00 precision and 0.59 recall when compared to human interpretations.

This method fails to extract semantic relations if the information type does not match a rule. Therefore, we propose a neural network approach to addresses this generalization problem which is not reliant on handcrafted semantic rules extracted from grounded analysis of information types.

### 3.3 Mobile App User Interface Analysis

To identify the trace links between privacy policy requirements and app code, we need to extract the user-provided information types from user interface (UI) input fields. For this reason, we analyzed 53 input fields from 19 apps available in Google Play [27]. We utilized crowdsourcing and free listing survey design [2] to elicit information types associated with each input field presented in screenshots. We recruited 30 participants per survey using Amazon Mechanical Turk that were located in the United States with an overall HIT approval rating greater than 95%.

Through the study, we obtained 30 information types per input field. Since there are multiple ways to describe the same concept, we pre-processed the results to more easily comparable elicited types [20, 21]. After pre-processing, we combined similar type names for each field and calculate the type name frequency, which is the number of workers who provided each syntactically unique type name per field. Finally, for each field, we selected the most frequent type name, which remains linked to a set containing the less frequent type names for that field.

We also inferred information types for the same 53 input fields by concatenating the file name and input field labels. The results were compared with the most frequent input types provided by crowd workers showing 33.9% match [27]. This suggest that a naive approach with local context is not effective. However, considering the number of mobile apps available in the market and their input fields, our current approach cannot be scaled. Therefore, we propose an automated approach which is discussed in Section 4.2.

## 4 PROPOSED WORK

Our proposed models and evaluation plans are presented in this section.

### 4.1 Automated Ontology Construction

To address the scalability and coverage problems in ontology construction approaches, we propose an automated approach for identifying relationships between two information types in privacy policies. This approach takes two inputs: "information type$_{LHS}$" and "information type$_{RHS}$" for left-hand side and right-hand side information types, respectively. Each word in the information type is represented using word embedding. The word embeddings are pre-trained vectors from the privacy policy domain. These embeddings are then fed into a convolutional neural network (CNN) component with four different convolutional filters [26] to ensure that we capture the local semantics of unigrams, bigrams, trigrams, and four-grams in the information types. We present the output of information type modeling components as two semantic vectors. Finally, we compare the similarity of the two semantic vectors and identify the relation between them using a softmax classifier. This architecture does not require any complicated syntax or semantic pre-processing (e.g. identifying the part of speech for each word in the information types) for inputs.

**Proposed Evaluation:** The proposed model requires training and testing datasets that include the relations assigned to information type pairs from a given lexicon. We plan to use the manually constructed ontology from 356 information types and 50 privacy policies mentioned in Section 3.1. This ontology contains 1,583 hypernymy and 310 synonymy relations. We also plan to use the information type pairs that are considered as unrelated in the ontology as part of our training set.

### 4.2 Automated User Interface Analysis

This section presents the overview of our proposed approach to elicit information types associated with user interface (UI) input fields [15]. Our proposed approach is based on the assumption on the naturalness of Android XML layout code, so that it is possible to directly apply natural language processing techniques to the layout

[1] http://polidroid.org/downloads/ontology.owl
[2] https://sites.google.com/site/uiprivacy2017/



RIGHTSLINK

code and extract the information types. This approach contains two main steps: (1) given a mobile app decompiled code, the UI analysis extracts the layout XML code and constructs a *context sequence* for each input field (`EditText`) which includes the id, text, and hint attributes of the `EditText`, and the id, text, and hint attributes of all views preceding the `EditText` in the layout XML file; (2) The sequence to sequence learning component takes an input field context sequence and maps it to a target sequence of words representing an information type phrase.

Information type phrases are comprised of sequence of words with various lengths that are not known at the time. To identify the information types from the UI context sequences, we plan to use two Long Short-term Memory (LSTM) networks [25] to map a source to a target sequence. First, we encode the input sequence to a vector of fixed dimension that includes the semantics of the input sequence using a multi-layered LSTM. Next, we feed the input vector to another LSTM which decodes the target sequence from the vector. The target sequence cannot be identified using a classification model since information types related to UI input fields are not bound to a finite set of well-defined phrases.

**Proposed Evaluation:** To train and evaluate the proposed model, we have acquired information types for 53 input fields as mentioned in Section 3.3. However, we understand that our current data does not contain sufficient amount of sequence pairs for training the model. We are planning to publish a new study to acquire additional training samples.

### 4.3 Mapping Construction

To identify trace links and detect potential violations, we plan to create a mapping from information types associated with input fields and privacy policy ontology concepts. For this reason, we plan to utilize the trained model proposed in Section 4.1. Given a pair of information types from Android layout code and privacy ontology, we can identify the semantic relation between them. If a UI input field information type can be mapped to an ontology phrase, but the phrase (or its synonyms in the ontology) does not exist in the privacy policy, a violation is detected. Furthermore, if some of the phrase's hypernyms do exist in the privacy policy, the violation is weak, while if none of the phrase's hypernyms exist in the policy, the violation is strong.

**Proposed Evaluation:** We plan to use the automatically constructed mapping in violation detection framework proposed by Wang et al. [27] and compare the violations identified through our mapping with the results reported for 80 popular finance and health apps from Google Play[27].

## ACKNOWLEDGMENTS

I deeply thank my Advisor Dr. Jianwei Niu, my thesis committee Xiaoyin Wang, Ravi Sandhu, and John Quarles from UTSA and Travis D. Breaux from CMU for helpful supervision, suggestion, and discussion.

## REFERENCES

[1] Annie I Anton and Julia B Earp. 2004. A requirements taxonomy for reducing web site privacy vulnerabilities. *RE* 9, 3 (2004), 169–185.
[2] Harvey Russell Bernard. 2011. *Research methods in anthropology: Qualitative and quantitative approaches.* Rowman Altamira.
[3] Jaspreet Bhatia, Morgan C Evans, Sudarshan Wadkar, and Travis D Breaux. 2016. Automated extraction of regulated information types using hyponymy relations. In *REW*. 19–25.
[4] Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT/EMNLP*. 724–731.
[5] Federal Trade Commission. 2010. Federal Trade Commission Act. Public Law 111-203.
[6] Juliet Corbin, Anselm Strauss, and Anselm L Strauss. 2014. *Basics of qualitative research.* Sage.
[7] European Parliament and Council. 2016. General Data Protection Regulation. Regulation (EU) 2016/679.
[8] Morgan C Evans, Jaspreet Bhatia, Sudarshan Wadkar, and Travis D Breaux. 2017. An Evaluation of Constituency-based Hyponymy Extraction from Privacy Policies. In *RE*. 312–321.
[9] Dieter Fensel. [n. d.]. Ontologies: A silver bullet for knowledge management and electronic-commerce (2000). *Berlin: Spring-Verlag* 143 ([n. d.]).
[10] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *ACL*. 427–434.
[11] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. 2018. Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning. *arXiv preprint arXiv:1802.02561* (2018).
[12] Kamala D Harris. 2013. *Privacy on the go: recommendations for the mobile ecosystem.*
[13] Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conf. on Computational linguistics-Volume 2*. 539–545.
[14] Mitra Bokaei Hosseini, Travis D Breaux, and Jianwei Niu. 2018. Inferring Ontology Fragments from Semantic Role Typing of Lexical Variants. In *REFSQ*. 39–56.
[15] Mitra Bokaei Hosseini, Xue Qin, Xiaoyin Wang, and Jianwei Niu. 2018. Extracting Information Types from Android Layout Code Using Sequence to Sequence Learning. In *Statistical Modeling of Natural Software Corpora Workshop of the Thirty-Second AAAI Conference on Artificial Intelligence*.
[16] Mitra Bokaei Hosseini, Sudarshan Wadkar, Travis D Breaux, and Jianwei Niu. 2016. Lexical Similarity of Information Type Hypernyms, Meronyms and Synonyms in Privacy Policies. In *2016 AAAI Fall Symposium Series*.
[17] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.
[18] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*. 1003–1011.
[19] Gabriele Petronella. 2014. Analyzing privacy of Android applications. (2014).
[20] Martin F Porter. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.
[21] Martin F Porter. 2001. Snowball: A language for stemming algorithms.
[22] Johnny Saldaña. 2015. *The coding manual for qualitative researchers.* Sage.
[23] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D. Breaux, and Jianwei Niu. 2016. Toward a Framework for Detecting Privacy Policy Violations in Android Application Code. In *ICSE*. 25–36.
[24] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*. 1297–1304.
[25] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.
[26] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*. 1422–1432.
[27] Xiaoyin Wang, Xue Qin, Mitra Bokaei Hosseini, Rocky Slavin, Travis D Breaux, and Jianwei Niu. 2018. Guileak: Tracing privacy policy claims on user input data for android applications. In *Proceedings of the 40th ICSE*. ACM, 37–47.
[28] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N Cameron Russell, et al. 2016. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of ACL (Volume 1: Long Papers)*, Vol. 1. 1330–1340.
[29] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research* 3, Feb (2003), 1083–1106.
[30] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation Classification via Convolutional Deep Neural Network.. In *COLING*. 2335–2344.
[31] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *ACL*, Vol. 2. 207–212.
[32] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven M. Bellovin, and Joel Reidenberg. 2017. Automated Analysis of Privacy Requirements for Mobile Apps. In *NDSS*.



RIGHTSLINK